Overcoming Occlusion with Inverse Graphics

Overcoming Occlusion with Inverse Graphics.

Pol Moreno[1], Christopher K.I. Williams[1], Charlie Nash[1], Pushmeet Kohli[2]

[1] School of Informatics, University of Edinburgh
p.moreno-comellas@sms.ed.ac.uk,ckiw@inf.ed.ac.uk, charlie.nash@ed.ac.uk
[2] Microsoft Research
pkohli@microsoft.com

Abstract. Scene understanding tasks such as the prediction of object pose, shape, appearance and illumination are hampered by the occlusions often found in images. We propose a vision-as-inverse-graphics approach to handle these occlusions by making use of a graphics renderer in combination with a robust generative model (GM). Since searching over scene factors to obtain the best match for an image is very inefficient, we make use of a *recognition model* (RM) trained on synthetic data to initialize the search. This paper addresses two issues: (i) We study how the inferences are affected by the degree of occlusion of the foreground object, and show that a robust GM which includes an outlier model to account for occlusions works significantly better than a non-robust model. (ii) We characterize the performance of the RM and the gains that can be made by refining the search using the GM, using a new dataset that includes background clutter and occlusions. We find that pose and shape are predicted very well by the RM, but appearance and especially illumination less so. However, accuracy on these latter two factors can be clearly improved with the generative model.

Keywords: Vision-as-inverse-graphics, scene understanding, occlusion

1 Introduction

Computer vision is fundamentally an ill-posed and extremely complex problem: there are many different scene configurations that can produce a given image, and many factors of variability. An old idea is to use a model of how images are generated to solve the inverse process, which can also be seen as an instance of analysis-by-synthesis, see e.g. [32]. In this work we make use of this idea, which we call vision-as-inverse-graphics, in order to extract detailed descriptions of an object in an indoor scene. Examples of these descriptions (or factors) include the object's shape and appearance, pose, and the illumination of the scene. Stevens and Beveridge [27] is an early example of combining vision and graphics.

Inverse-graphics is an elegant solution in which these factors are used as the variables that explain the generation of images. However, searching for the descriptions that best explain an image is a challenging task due to their high dimensionality, thus we make use of discriminative models, which we will call recognition models (RMs), as a way of cutting down the search (see e.g. Dayan

et al. [6] and Williams et al. [28]). Combining the bottom-up and top-down information flows is a key aspect in the design of our solution, see Figure 1. Our goal is to extract fine-grained descriptions that can be used for many different tasks, such as robotic systems that need to interact with the world.

Recent work (see e.g. Yildirim et al. [31]) has made some exploration of the combination of generative and recognition models for scene understanding. However, the experiments to date have typically been made on "clean" scenes which do not contain occluding objects and background clutter. Our contributions are:

- We study how the inferences are affected by the degree of occlusion of the foreground object, and show that a robust generative model which includes an outlier model to account for occlusions works significantly better than a non-robust Gaussian model.
- We characterize the performance of the recognition model and the gains that can be made by refining the search using a generative model. We find that pose and shape are predicted very well by the RM, but appearance and especially illumination less so. However, accuracy on these latter two factors can be clearly improved with the generative model.
- Production of a new synthetic dataset with which one can evaluate the performance of the models with variation across pose, shape, appearance, complex illumination (extracted from a collection of indoor environment maps), a diverse set of indoor scene backgrounds, and with the foreground object being partially occluded. This goes beyond prior work (e.g. [31]) which only explores shape, appearance and limited lighting variation.

The structure of the paper is as follows: we first describe the **generative model** as a differentiable renderer (see section 2). The idea is to make use of approximate gradients that describe how the image intensities change with respect to the scene factors, leading to efficient search over these. The robust and Gaussian likelihood models are also explained here. The **recognition model** (see section 3) is trained discriminatively to predict the scene factors of interest. In order to do this we have created a **synthetic dataset** of indoor scenes in which we can generate novel instantiations of object classes of interest, as explained in section 4. The experimental setup and results are given in section 5.

1.1 Related work

Gradient based approaches. The problem of inferring the physical world that gave rise to a given image is a long-standing and fundamental problem in computer vision. One recent example of this reconstructive paradigm is the work of Barron and Malik [1] where depth maps, reflectance maps, and illumination models are recovered from an input image by optimizing a data fit criterion and making use of prior distributions. Another nice example is the work of Loper and Black [21] as applied to the problem of fitting an articulated human body model to Kinect data.

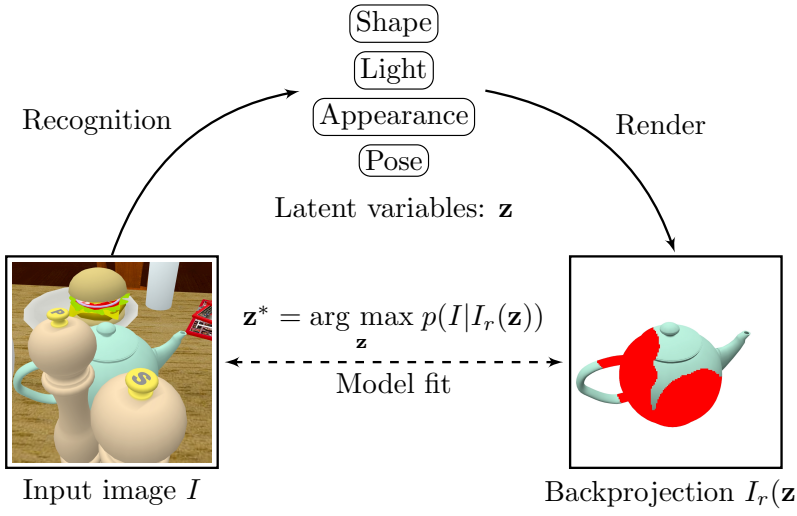


Fig. 1: Illustration of our inverse-graphics solution. Given an image, the recognition models of shape, lighting, appearance and pose initialize the latent variables. Then, the differentiable renderer generates an image which is fitted by optimizing over the latent variables $\mathbf{z}$ that best match the input image, i.e. $\mathbf{z}^* = \arg\max_{\mathbf{z}} p(I_{GT}|I_r(\mathbf{z}))$. Detected occlusion mask is illustrated with red

Hybrid generative-discriminative architectures. In the above papers the search over the latent factors of variation is made directly using gradient-based optimization methods. However, it is also possible to make use of a recognition model to predict these factors bottom-up. One example is the algorithm behind Kinect [26], which is shown to be a key component of efficiently estimating human pose.

The work of Jampani et al. [15] and Kulkarni et al. [18] makes use of bottom-up sampling probability distribution proposals to get an approximation of the posterior probability of the latent variables more efficiently for black-box renderers. Both papers show improved sampling performance using these data-driven proposals.

Krull et al. [17] work on estimating 6D pose from RGB-D images. Rather than explicitly modelling occlusions they learn an energy function that compares the rendered model and the input image (which may contain occlusions) using a convolutional neural network. At inference time a search is carried out in 6D pose space to minimize the energy function. Note that this work is restricted to inference of pose, while we also address object shape, appearance and illumination factors.

Perhaps the most closely related work to ours is that of Yildirim et al. [31] which makes use of both a recognition model and a generative model. They use the morphable face model [5] and explore variation in shape, texture, pose,

and lighting direction. In our work we make use of more complex illumination (extracted from a collection of indoor environment maps), and also include scene backgrounds and foreground occlusions. We also analyze the performance of the recognition model with respect to the different factors. A further difference is that Yildirim et al. [31] make use of MCMC methods for inference, while we use local search using a differentiable renderer.

The Picture framework of [20] provides a general probabilistic programming framework for specifying scene understanding problems and general-purpose inference machinery, and applies it to a number of example problems. Rather than generating pixels it uses a "representation layer" of features to assess the match of an input image and the rendered scene, using a likelihood-free method. Note, however, that this means there is a choice of representation that needs to be made (by hand) in order to use the framework. Also, Picture does not explicitly handle occlusions, nor complex lighting.

Learning the generative model. Another piece of related work is to use a deep learning auto-encoder model for both the recognition model (encoder) and generative model (decoder), as in Kulkarni et al. [19]. This DC-IGN model does not make use of an explicit 3-d graphics model, but rather learns a mapping from the graphics code layer (hidden units) to images in the decoder. This gives a lot of freedom to the model as to how it wishes to represent variation in shape, illumination etc. However, it also means that the effect of e.g. pose variation has to be learned specifically for an object class, rather than being generic geometric knowledge. The ability of DC-IGN to generalize to novel images (e.g. under different rotations) seems to be limited if one looks at the visual quality of its renders. Experiments are again conducted wrt shape, texture, pose, and simple lighting direction variation, without scene backgrounds and foreground occlusions. Note also that the auto-encoder architecture means that there is no scope for refinement of the code predicted by the recognition model, in contrast to our work and that of [31].

The work of Dosovitskiy et al. [8] is somewhat similar to the DC-IGN model in that it learns a "decoder" network from variables representing shape class, pose and other transformations (e.g. in-plane rotation, colour transformations), although there is no corresponding encoder network. Their network shows impressive generalization over the chair class, but is not used to address issues of scene understanding.

2 Generative models

We first describe how the foreground object is modelled, and then discuss the rendering process and our likelihood models.

2.1 Object model

The factors that characterize a visual object are the pose, shape, appearance, and the incident illumination. The choice parametrization is a trade-off between

its ability to model complex scenes and having a compact representation for efficient search over the latent variables. In this work we use the following:
Shape: We model the shape of a given object class using a deformable mesh characterized by Principal Component Analysis, as used e.g. by Blanz and Vetter [5] for modelling faces.
Pose (2D): We assume the camera is centered on the foreground object at a fixed distance with variable azimuth and elevation. Note that, to a large degree, the shape parameters can model the general scale of our object, thus including a distance variable for the position of the camera is not necessary.
Appearance (3D): A global RGB colour is used for all object vertices, which we also refer to as the albedo. The reflectance function is assumed to be Lambertian, as explained in more detail below.
Illumination (9D): Inverse illumination is known to be an ill-posed problem [22, Ch. 6], with the additional issue that real illumination can be very complex (thus modelled poorly by single directional light sources). We are interested in representations that are practical for our inverse-graphics framework. One of the most natural representations is to project lighting to basis functions such as Spherical Harmonics, see e.g. [23], a technique which is also widely used in computer graphics for efficient approximate rendering of lighting. Other representations include Haar wavelets [25], and so called basis images of the object class illuminated from different light conditions [3]. We use Spherical Harmonics (SH) due to its compact representation and the efficiency at which re-lighting is computed when rotating the light or changing the shape and pose of the modelled objects.

Spherical Harmonics form an orthogonal basis that can be used to approximate complex illumination. For Lambertian reflectance, it can be shown [2] that only 9 components can approximate images of a convex object by at least 99.22%. Lambertian reflectance is a diffuse surface property where the resulting RGB reflectance $\mathbf{r}(\mathbf{x}_i)$ of an incoming light a point $\mathbf{x}_i$ on the surface is given by

$$\mathbf{r}(\mathbf{x}_i) = \mathbf{a}_i \max(\mathbf{n}_i \cdot \mathbf{l_i}, 0), \tag{1}$$

where $\mathbf{a}_i$ is the RGB albedo, $\mathbf{l}_i$ is the incoming light direction and $\mathbf{n}_i$ is the vertex normal at point $\mathbf{x}_i$. Even though real world objects have cast shadows, specularities, and other non-Lambertian factors, our assumption is that a Lambertian approximation is sufficient for a large variety of tasks in inverse-graphics.

2.2 Renderer

We use a graphics renderer as a generative model that takes as input a set of object meshes (which include information of their vertex coordinates, normal vectors, textures and colors) and then generates the renders using OpenGL. Our renderer is based on OpenDR: Differentiable Renderer [21], which we have extended to make it capable of rendering multiple objects and textures, and also modernized its OpenGL back-end in order to support modern graphics features and hardware-accelerated rendering. The main advantage of OpenDR

is that it provides approximate derivatives of the image with respect to the latent variables or any variable of interest. The approximation stems from the fact that the rendering function is non-differentiable due to self-occlusion and occlusion across objects. Having derivatives plays a key role in the efficiency of the optimization process as the dimensionality of our latent space increases

We can think of the cost function of our generative process (i.e. the reconstruction error) in terms of a likelihood function. In this work we explore two possible models over the pixel intensities. Since we are modelling one foreground object, all the rendered pixels that lie outside the object mask are are considered as background and modelled by a uniform distribution. Given an image I, the likelihood of the foreground pixel (fg) intensities are modelled as follows:

- **Gaussian model:** The simplest case is a Gaussian distribution on the pixel intensities:

$$p(\mathbf{c}_i|\mathbf{z}, fg) = \mathcal{N}(\mathbf{c}_i; \boldsymbol{\mu}_i(\mathbf{z}), \sigma^2 I) \tag{2}$$

  where $\mathbf{c}_i$ is the RGB color at pixel i, $\boldsymbol{\mu}_i(\mathbf{z})$ is the RGB color output of the renderer given scene latent variables $\mathbf{z}$, σ^2 is the spherical variance (assumed to be the same for all pixels), and $\mathcal{N}$ denotes the Gaussian distribution. In terms of the optimization landscape, this is equivalent to using a squared error cost function.

- **Robust model:** We want our model to tolerate foreground occlusions by using outlier statistics as in Williams and Titsias [29]

$$p(\mathbf{c}_i|\mathbf{z}, fg) = \alpha\mathcal{N}(\mathbf{c}_i; \mu_i(\mathbf{z}), \sigma^2 I) + (1-\alpha)\mathcal{U}(\mathbf{c}_i)), \tag{3}$$

  where α is the mixing probability of a pixel being unoccluded, and $\mathcal{U}$ is the uniform distribution. Note that we can learn α from our training set by e.g. taking the average of the proportion of unoccluded pixels.

Thus, the overall log-likelihood of an image given the scene latent variables $\mathbf{z}$ is given by

$$L = \sum_{i \in fg} \log p(\mathbf{c}_i|\mathbf{z}, fg) + \sum_{j \in bg} \log \mathcal{U}(\mathbf{c}_j). \tag{4}$$

The pixel-wise outlier model as used above does not impose spatial priors on regions of occlusion. One could enhance this e.g. using Markov random field models on the occlusion labels (see e.g. [11]), at the cost of greater complexity in inference. One could also consider more complex occlusion models that learn the structure of occlusions from data, see e.g. [12]. While Black and Rangarajan [4] propose robust statistical techniques for a number of vision tasks, the advantage of the latent variable formulation of our robust model is that it allows us to explain occlusions by using the posterior probabilities of the foreground/outlier pixel segmentation.

Estimating the latent variables of interest is obtained by initializing the OpenDR input with the estimates of the latent variables given by the recognition model, followed by locally improving its fit using the likelihood function,

e.q. 4, subject to the constraint that the PCA coefficients of the shape model lie within ± 3 standard deviations of the mean. This is why it is essential that the predictions of the recognition model should lie within the basin of attraction of the generative model.

3 Recognition model

The choice of the recognition models architecture depends on the task domain and the scene factors we want to infer. For instance, the model can output a single point estimate or take a probabilistic approach; in this work we focus on the former case. Again, we want the estimates of the bottom-up predictions to lie within the basin of attraction of the generative model.

Using the right feature representation of the images is also key to good discriminative performance. After experimenting with a diverse set of features for each type of parameter (e.g. Histogram of Gradients for pose prediction, and different basis expansions for illumination) we found that learning the features from raw images gave the best prediction performance. Therefore, we make use of Convolutional Neural Networks (CNNs) to predict each of the latent variables as these have been key to the success for many computer vision tasks in recent years [16]. The architecture of choice is almost the same in all cases as it showed a good generalization capability: three convolutional layers (64 5x5 filters) with max-pooling and two dense hidden layer (with 256 and 32 hidden units each). In the case of pose and shape CNNs, the input is assumed to be grayscale. A dropout rate of 0.5 was used with Nesterov Accelerated Gradient descent with 0.9 momentum.

4 Synthetic ground-truth dataset

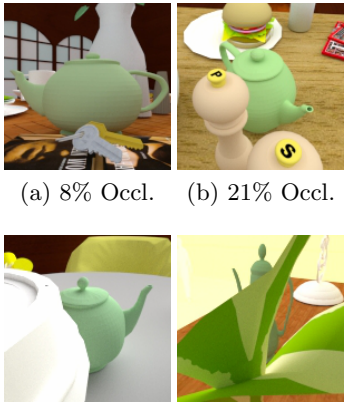
Synthetic datasets have been used for a variety of computer vision tasks, see e.g. [26, 33]. In order to generate synthetic data for training and evaluation of our method, we use an indoor scene generator based on the dataset of Fisher et al. [10]. This consists of 10,000 CAD models they collected and stochastically arranged together in different plausible indoor scenes. Here, we use over 80 different indoor scenes.

In this work we focus on the teapot object class. The reason for choosing teapots is that they are a common object in indoor scenes and have a good degree of variability. However, our stochastic scene generation is not limited to only one type of object, and many different object models can be collected from sources such as the Princeton Modelnet [30] and easily embedded into the scenes. 10 PCA dimensions were chosen so as to capture 90% of the variation in the training dataset of 23 3D teapot models.

To generate the scenes with complex and varied illumination settings, we collected over 70 complex indoor environment maps (also known as light probes) from different sources, see e.g. [7]. An environment map is the projection of the incident illumination on a point of a scene into e.g. an equirrectangular

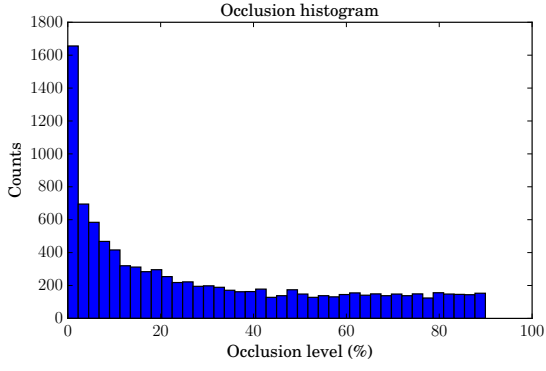
image. We use these environment maps to render illumination using Spherical Harmonics as in [24]. Since the Lambertian reflectance function acts as a low-pass filter, it effectively removes high-frequency information [2], hence the resulting illuminated objects are perceptually well approximated compared to using the actual environment maps.

The ground-truth of each sample in the dataset is generated by instantiating the teapot object in one of the indoor scenes. This involves sampling the shape parameters from the shape prior, as well as a uniformly sampled object rotation around the Z-axis (up axis). The camera's angles of azimuth and elevation are randomly sampled with a uniform distribution on the upper hemisphere and it is placed at a fixed distance from the object. The scene is then illuminated by one of the environment maps, rotated uniformly along the Z-axis. In total, we produced 10,000 training and 1,000 test images by the above process. We distinguish two different types of rendering methods: a non-photorealistic rendering which does not include global illumination and uses OpenGL, and a photorealistic rendering which uses the unbiased ray-tracer Cycles[1]. The OpenGL rendering is much quicker and we use it to generate the training data. Their width and height are set to 150 by 150 pixels. Figure 2 shows a few examples rendered with Cycles. Figure 2 (c) shows the histogram of occlusion levels in our dataset, note how there are occlusions of all levels up to 90% (it is not useful to have images with occlusions near 100% for the purpose of training or evaluation). This scene generator and dataset will be made available upon the publication of this work.



(a) 8% Occl. (b) 21% Occl.

(d) 39% Occl. (e) 70% Occl.



(c) Histogram of occlusions

Fig. 2: Examples of our synthetic data-set with different levels of occlusion

[1] http://www.blender.org/manual/render/cycles/introduction.html

5 Experiments

5.1 Experimental setup

In the following experiment we use a test set which consists of 1000 test samples from our synthetic dataset with different levels of occlusions. These test images are rendered with the unbiased ray-tracer Cycles in order to assess our models using scenes with global illumination which simulate real images illumination. We are interested in understanding the performance of our models (recognition, gaussian and robustified) as the occlusion level is increased. As a reference evaluation, we provide a baseline prediction based on the mean values of the different factors on the training set (mean baseline). The evaluation metrics and optimization procedure for this experiment are explained below.

Evaluation metrics. In general, evaluating predictions of the fine-grained scene latent variables (e.g. pose, shape, appearance, illumination) is difficult due to a lack of labelled datasets, but here we have the advantage of having a synthetic dataset in which all these factors are designed to have a rich variability. One way to evaluate the performance of our method is the log-likelihood on the test set. However, that does not give us an easily interpretable quantity. Instead, evaluation is carried out with respect to the predictive performance of the latent variables we are modelling, which have a more intuitive and physical interpretation. We also evaluate how well the unoccluded foreground pixels are predicted.

Choosing the evaluation metric for each of these factors in not necessarily straightforward. For pose azimuth and elevation, we define the errors to be the absolute angular difference for each angle (azimuth and elevation variables range 360 and 90 degrees respectively). Special care needs to be taken to measure the appearance and illumination attributes as they interact with each other multiplicatively, see eq. 1. For the appearance error e_a^i, we convert the colour representation to Lab space[2] which is more perceptually uniform, and omit the luminance L to give

$$e_a^i = \sqrt{(a_{pred}^i - a_{gt}^i)^2 + (b_{pred}^i - b_{gt}^i)^2}, \tag{5}$$

where a_{pred}^i and b_{pred}^i correspond to the predicted color dimensions of the Lab representation, and a_{gt}^i and b_{gt}^i to the ground-truth values for test image i.

For the illumination error we use the mean squared error (MSE) of the Spherical Harmonics coefficients. It is known that estimating the illumination from a single view can be potentially ill-conditioned [22, Chapter 6] so it is possible that the ground-truth illumination cannot be recovered exactly. It is easy to see that our metric is equivalent to the MSE of the incident SH illumination projected on a sphere. Furthermore, we use a scale-invariant version of this metric in order to account for the fact that appearance and illumination interact with each other

[2] https://en.wikipedia.org/wiki/Lab_color_space

multiplicatively. A similar evaluation metric is used in [1]. Finally, we evaluate shape reconstruction by using the mean Euclidean distance between the vertices of the ground-truth and predicted meshes, both aligned to a canonical pose.

The occlusion predictions from the robust model are evaluated thus: we obtain predicted un-occluded foreground pixels by evaluating the posterior probability at each pixel of belonging to the foreground or outlier component in eq. 3, and thresholding at 0.5. These are then compared to the ground-truth unoccluded foreground pixels using the segmentation accuracy as defined e.g. in [9], where the number of true positives is divided by the sum of the true positives plus false positives plus false negatives. For the recognition model without iterative refinement, we assume that all of the predicted foreground pixels are unoccluded.

Optimization procedure. For the robust model we explored different optimization strategies since convergence is sensitive to the choice of the likelihood variance. If we use a pixel variance that is too large, then occlusions and background clutter will affect the optimization negatively. On the other hand, if the variance is too small, the robust model tends to ignore large parts of the image including those which are important for a correct optimization. We jointly optimize the latent variables of pose, shape, appearance and illumination using a standard deviation of $\sigma = 0.03$ of the pixel likelihood. Note that the pixel color for each RGB channel ranges between 0 and 1. Also, we clamp shape parameters to be less than three standard deviations from the prior mean: it is much more likely that the optimization has gone wrong than it is having an input image with an unreasonably large (or small) deformation of the mesh.

We explored different minimization methods including gradient descent (with and without momentum and decay), nonlinear conjugate gradient (CG)[3], dogleg CG (as used in the OpenDR work), and BFGS. Nonlinear CG consistently converged to better minima in our tests hence we use it in our experiments. It is not surprising that these methods often converge differently since the optimization landscape is non-convex and the derivatives are approximate.

5.2 Results

Effect of Occlusion on accuracy. Fig. 3 shows the median cumulative predictive performance as a function of the percentage occlusion when evaluating on the photorealistic test images. (So, for example the performance at 75% occlusion is obtained from all test cases with this much occlusion or less.) Note that in panels (a)-(e) lower error is better, while in panel (f) higher scores are better. Comparing the recognition network to the recognition network plus robust fitting, we see very similar performance for azimuth, elevation and shape factors, but marked improvements with fitting for appearance, illumination and occlusion. Pose and shape are predicted very well by the recognition model, which suggests that it is good at inferring latent variables for which there are clear and

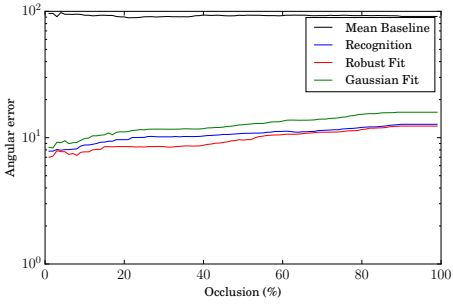
[3] http://learning.eng.cam.ac.uk/carl/code/minimize/

localized cues (e.g. for pose, the locations of tip of the spout, the handle, etc.), even under high levels of occlusion. Subsequent fitting of azimuth and shape factors on average shows little to no improvement, but we do see an improvement for elevation. We also see how the recognition model has not learned to predict the illumination and that occlusion has a strong effect on the prediction. This agrees with the intuition that the other factors can be more easily inferred by having a smaller portion of the object unoccluded.

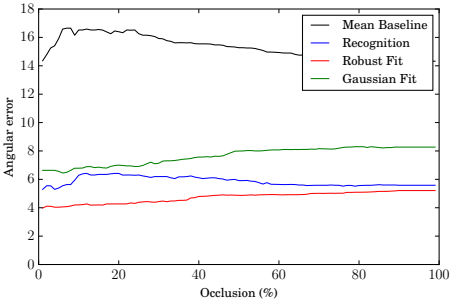
Fitting results. It is noticeable that the Gaussian model performs much worse than the robust model for elevation, appearance, and illumination. Indeed it sometimes performs worse after fitting than the recognition model itself, which is most likely explained by the fact that the Gaussian model does not handle occlusion and background clutter well. Plots of the mean performance rather than the median show similar trends to Fig. 3, except that the elevation angular error of the robust fit becomes worse than the recognition model over the occlusion range (increasing from 6 to 8 degrees). Analysis shows that this is due to some large errors that arise particularly at higher occlusion levels. A possible remedy we are investigating is to check if the fitting process has gone awry, and if so revert to the recognition model prediction. Table 1 summarizes the plots by showing the median prediction errors for the baseline, the recognition model prediction, and the Gaussian and robust fits for up to 75% occlusion. For reference, we also show the results of the recognition and fitting when the test images are rendered with OpenGL, which is the method used for our differentiable renderer as well as to render the training images of the recognition models. We notice how the relative improvements when fitting is similar in both Cycles and OpenGL cases, but the OpenGL experiments have overall lower errors as is expected. Notice how robust fitting gives the best performance in all cases except for a slightly worse performance on elevation at high occlusion levels. We provide the median plots with both Cycles and OpenGL experiments in the supplementary materials along with some example videos of the fitting process.

Table 1: Median prediction errors for the latent variables for level of occlusion of 75%. Higher is better for segmentation evaluation

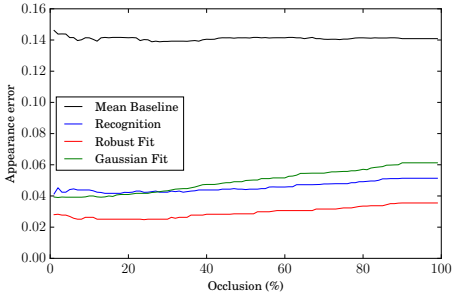
	Photorealistic				OpenGL		
	Baseline	RM	Gaussian	Robust	RM	Gaussian	Robust
Azimuth	92.14	11.71	14.52	**11.17**	10.68	17.18	**8.55**
Elevation	14.44	5.58	8.19	**5.02**	4.94	7.57	**3.668**
Appearance	0.140	0.048	0.056	**0.032**	0.036	0.055	**0.014**
Illumination	0.021	0.023	0.024	**0.019**	0.022	0.024	**0.019**
Shape	1.005	0.541	0.560	**0.511**	0.525	0.599	**0.479**
Segmentation	-	0.659	-	**0.778**	0.670	-	**0.828**



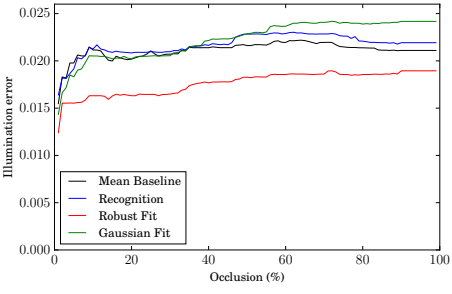
(a) Azimuth errors



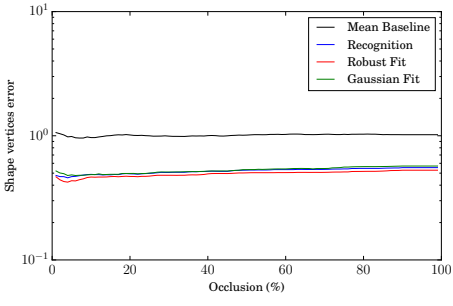
(b) Elevation errors



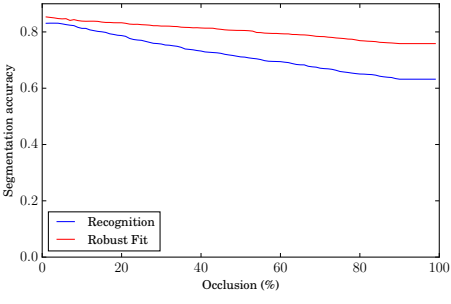
(c) Appearance errors



(d) Illumination errors



(e) Shape vertices errors



(f) Segmentation accuracy (higher is better)

Fig. 3: Median errors for azimuth, elevation, appearance, illumination, shape; figure e) is the segmentation accuracy

Qualitative evaluation. Fig. 4 shows examples of how the robust model explains away the occlusion rather impressively for different levels occlusion, which is something difficult to achieve with purely bottom-up techniques. In Fig. 5 we show a few examples to illustrate how the environment map illumination is fitted to capture the main directional sources of illumination in a complex indoor illumination setting with multiple light sources. Note that all the scene latent variables were predicted and fitted in this experiment. Indeed, the robust model seems much more capable of capturing fine-grained descriptions such as illumination, appearance and occlusions than the recognition model.

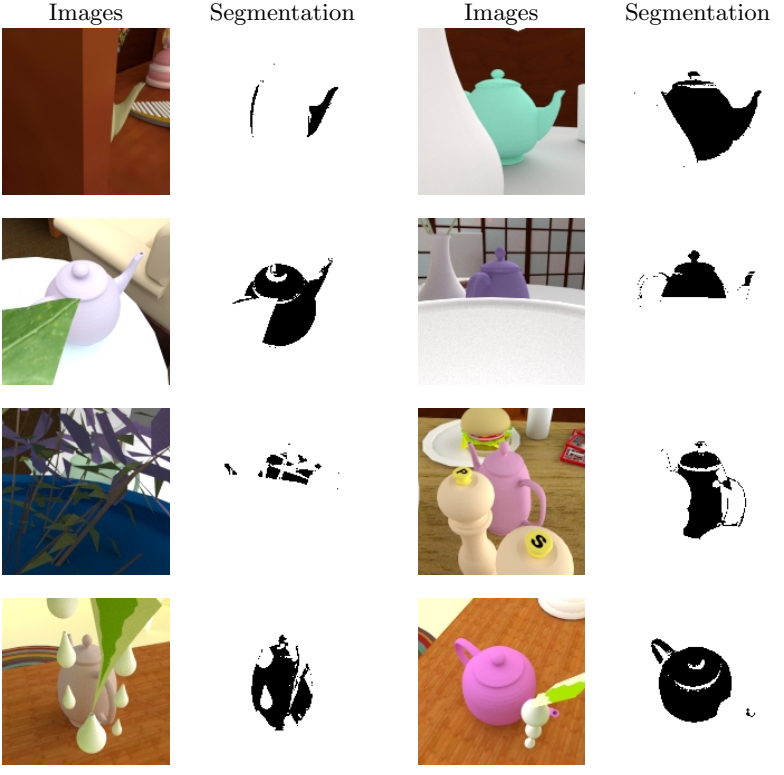


Fig. 4: Examples of occlusion inference using the posterior of the robust pixel probabilities

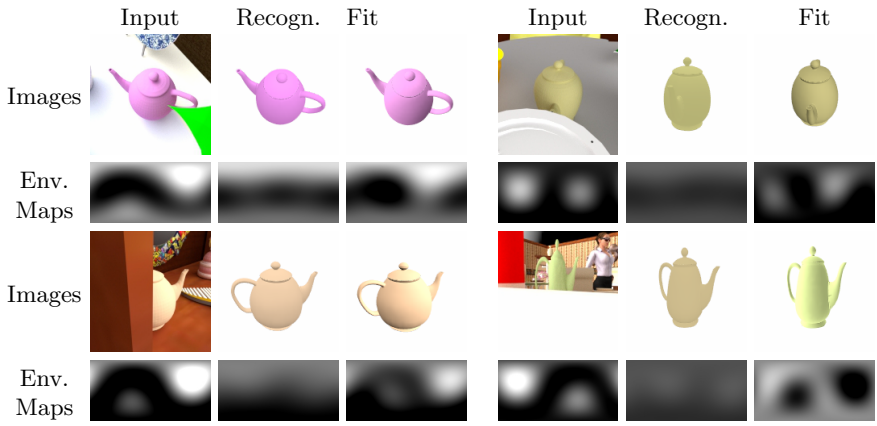


Fig. 5: Examples of how fitting with the robust model captures the correct illumination whereas RM is unable to do so

6 Discussion

Above we have investigated the use of vision-as-inverse-graphics with recognition models for a target object embedded in background clutter and subject to occlusions. A great advantage of using data generated by computer graphics is that it allows us complete access to the underlying scene parameters, and hence the ability to explore these systematically.

Our results show that some of the latent variables like pose and shape are well-predicted by the recognition network, while others such as illumination, appearance and occlusion benefit from subsequent refinement by fitting a robust generative model. Our results also show that the robustified model of e.q. 3 clearly outperforms a Gaussian likelihood, and provides a way to detect occlusions even in complicated cases.

Future directions for research include investigating: detecting if the fitting process has fallen into improbable basins of attraction, and the use of multimodal predictions in the recognition network (as per [14] or [13]).

References

1. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. IEEE Trans. Pattern Analysis and Machine Intelligence (2015)
2. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. Pattern Analysis and Machine Intelligence, IEEE Trans. 25(2), 218–233 (2003)
3. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions? International Journal of Computer Vision 28(3), 245–260 (1998)
4. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision 19(1), 57–91 (1996)

5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer graphics and Interactive Techniques. pp. 187–194 (1999)
6. Dayan, P., Hinton, G.E., Neal, R.M., Zemel, R.S.: The Helmholtz Machine. Neural Computation 7(5), 889–904 (1995)
7. Debevec, P.: Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. pp. 189–198. SIGGRAPH '98, New York, NY, USA (1998)
8. Dosovitskiy, A., Springenberg, J.T., Brox, T.: Learning to generate chairs with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1538–1546 (2015)
9. Everingham, M., Gool, L.V., Williams, C.K.I., J. Winn, A.Z.: The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88(2), 303–338 (2010)
10. Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., Hanrahan, P.: Example-based synthesis of 3d object arrangements. ACM Transactions on Graphics (TOG) 31(6), 135 (2012)
11. Fransens, R., Strecha, C., Van Gool, L.: A Mean Field EM-algorithm for Coherent Occlusion Handling in MAP-Estimation Problems. In: Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on (2006)
12. Gao, T., Packer, B., Koller, D.: A segmentation-aware object detection model with occlusion handling. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on (2011)
13. Guzman-Rivera, A., Kohli, P., Glocker, B., Shotton, J., Sharp, T., Fitzgibbon, A., Izadi, S.: Multi-Output Learning for Camera Relocalization. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. pp. 1114–1121. IEEE (2014)
14. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive Mixtures of Local Experts. Neural Computation 3, 79–87 (1991)
15. Jampani, V., Nowozin, S., Loper, M., Gehler, P.V.: The informed sampler: A discriminative approach to Bayesian inference in generative computer vision models. Computer Vision and Image Understanding 136, 32–44 (2015)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems 25 pp. 1–9 (2012)
17. Krull, A., Brachmann, E., Michel, F., Ying Yang, M., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 954–962 (2015)
18. Kulkarni, T.D., Mansinghka, V.K., Kohli, P., Tenenbaum, J.B.: Inverse Graphics with Probabilistic CAD Models. arXiv preprint arXiv:1407.1339 (2014)
19. Kulkarni, T.D., Whitney, W., Kohli, P., Tenenbaum, J.B.: Deep Convolutional Inverse Graphics Network. Neural Information Processing Systems (NIPS) (2015)
20. Kulkarni, T.D., Kohli, P., Tenenbaum, J.B., Mansinghka, V.K.: Picture: a probabilistic programming language for scene perception. Computer Vision and Pattern Recognition, 2015. CVPR 2015. IEEE Conference on (2015)
21. Loper, M.M., Black, M.J.: OpenDR: An Approximate Differentiable Renderer. In: Computer Vision–ECCV 2014, pp. 154–169. Springer (2014)

22. Ramamoorthi, R.: A signal-processing framework for forward and inverse rendering. Ph.D. thesis, Stanford University (2002)
23. Ramamoorthi, R.: Modeling illumination variation with spherical harmonics. Face Processing: Advanced Modeling Methods pp. 385–424 (2006)
24. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 497–500. ACM (2001)
25. Reinhard, E., Heidrich, W., , P., Pattanaik, S., Ward, G., Myszkowski, K.: High dynamic range imaging: acquisition, display, and image-based lighting. Morgan Kaufmann (2010)
26. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM 56(1), 116–124 (2013)
27. Stevens, M.R., Beveridge, J.R.: Integrating Graphics and Vision for Object Recognition. Kluwer Academic Publishers, Boston (2001)
28. Williams, C.K.I., Revow, M., Hinton, G.E.: Instantiating deformable models with a neural net. Computer Vision and Image Understanding 68(1), 120–126 (1997)
29. Williams, C.K.I., Titsias, M.K.: Greedy learning of multiple objects in images using robust statistics and factorial learning. Neural Computation 16(5), 1039–1062 (2004)
30. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1912–1920 (2015)
31. Yildirim, I., Kulkarni, T.D., Freiwald, W.A., Tenenbaum, J.B.: Efficient analysis-by-synthesis in vision: A computational framework, behavioral tests, and comparison with neural representations. In: Proceedings of the Thirty-Seventh Annual Conference of the Cognitive Science Society (2015)
32. Yuille, A., Kersten, D.: Vision as Bayesian inference: analysis by synthesis? Trends in Cognitive Sciences 10(7), 301–308 (2006)
33. Zia, M.Z., Stark, M., Schiele, B., Schindler, K.: Detailed 3D Representations for Object Recognition and Modeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(11), 2608–2623 (2013)